EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	9	(exclusive$3 adj2 (control$4 or access$4)) with (queu$3) with (defer$4 delay$3 hold suspend$3 yield) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 09:56
L4	350	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 hold suspend$3 yield$3 block$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 10:11
L6	350	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 hold suspend$3 yield$3 block$3 postpon$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 10:40
L7	4	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 hold suspend$3 yield$3 block$3 postpon$3) with (other another) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 10:34
L9	5	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3 buffer) with (defer$4 delay$3 hold suspend$3 yield$3 block$3 postpon$3) with (other another) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 10:37
L11	11	dec same print with queu$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 10:39
L12	15	(fifo (first near in near out)) with exclusive with queu$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 10:44

EAST Search History

L15	99	(718/102-104.ccls. or 710/20,28, 36,39-40,54,200.ccls. or 709/225, 229.ccls.) and ((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 hold yield$3 postpon$3 pend$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 11:07
L16	46	(718/102-104.ccls. or 710/20,28, 36,39-40,54,200.ccls. or 709/225, 229.ccls.) and ((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 postpon$3 pend$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 11:07
L17	14	(718/102-104.ccls. or 710/20,28, 36,39-40,54,200.ccls. or 709/225, 229.ccls.) and ((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 postpon$3 stall$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 11:22
L19	5	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (defer$4 delay$3 postpon$3 stall$3) with (other another different) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:19
L21	1	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (job with queu$3) with (defer$4 delay$3 postpon$3 stall$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 11:50
L22	49	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (queu$3) with (high$4 increas$3 boost$3) with priorit$3 and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:02

EAST Search History

L23	119421	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (device peripheral printer) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:03
L24	182	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (device peripheral printer) same queu$3 and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:03
L25	42	((exclusive$3 adj2 (control$4 or access$4)) or lock) with (device peripheral printer) same queu$3 same (defer$4 delay$3 postpon$3 stall$3 priorit$5) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:04
L26	0	("2002/0083114").URPN.	USPAT	OR	OFF	2006/04/06 12:08
L27	5	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with (queu$3) with (defer$4 delay$3 postpon$3 stall$3) with (other another different) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:30
L34	63	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with (queu$3) with (defer$4 delay$3 postpon$3 stall$3) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:08
L35	58	l34 not l25	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 12:31

EAST Search History

L36	80	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with (queu$3) with ((defer$4 delay$3 postpon$3 stall$3 advanc$3) or ((increas$3 decreas$3) with priority)) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:19
L37	4	(task near2 schedul$3) same (lowering decreasing increasing near4 priority).ab. and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:13
L38	28	(task near2 schedul$3) same (lower$3 decreas$3 increas$3 near4 priority).ab. and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:13
L39	75	l36 not l25	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:14
L40	79	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with (queu$3) with ((defer$4 delay$3 postpon$3 stall$3 advanc$3) or ((increas$3 decreas$3) near3 priority)) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:38
L41	75	l36 not l25	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:19
L42	38	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with (queu$3) with ((advanc$3 ahead) or ((increas$3 decreas$3) near3 priority)) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 13:59

EAST Search History

L43	14	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with ((increas$3 decreas$3) near2 priority) and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 14:02
L45	11	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) with ((increas$3 decreas$3) near2 priority) and queu$3 and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 15:23
L46	38	((exclusive$3 adj2 (control$4 or access$4)) or lock or mutex or (mutual near exclusion)) same job near queu$3 and ((@ad<"20011113") or (@prad<"20011113") or (@rlad<"20011113"))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/04/06 15:23

Terms used **exclusive control defer printing** Found **33,979** of **173,942**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 An optimal algorithm for mutual exclusion in computer networks

Glenn Ricart, Ashok K. Agrawala

January 1981 **Communications of the ACM**, Volume 24 Issue 1

Publisher: ACM Press

Full text available: pdf(752.18 KB) Additional Information: full citation, references, citings, index terms

Keywords: concurrent programming, critical section, distributed algorithm, mutual exclusion, network, synchronization

2 A Majority consensus approach to concurrency control for multiple copy databases

Robert H. Thomas

June 1979 **ACM Transactions on Database Systems (TODS)**, Volume 4 Issue 2

Publisher: ACM Press

Full text available: pdf(2.32 MB) Additional Information: full citation, abstract, references, citings, index terms

A "majority consensus" algorithm which represents a new solution to the update synchronization problem for multiple copy databases is presented. The algorithm embodies distributed control and can function effectively in the presence of communication and database site outages. The correctness of the algorithm is demonstrated and the cost of using it is analyzed. Several examples that illustrate aspects of the algorithm operation are included in the Appendix.

Keywords: clock synchronization, computer networks, concurrency control, distributed computation, distributed control, distributed databases, multiprocess systems, update synchronization

3 Concurrency control: methods, performance, and analysis

Alexander Thomasian

March 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 1

Publisher: ACM Press

Full text available: pdf(427.18 KB) Additional Information: full citation, references, citings, index terms

Keywords: Markov chains, adaptive methods, concurrency control, data contention, deadlocks, flow diagrams, load control, optimistic concurrency control, queueing network models, restart-oriented locking methods, serialiazability, thrashing, two-phase locking,

two-phase processing, wait depth limited methods

4 Transactional lock-free execution of lock-based programs

Ravi Rajwar, James R. Goodman

October 2002 **ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 36 , 30 , 37 Issue 5 , 5 , 10

Publisher: ACM Press

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings

This paper is motivated by the difficulty in writing correct high-performance programs. Writing shared-memory multi-threaded programs imposes a complex trade-off between programming ease and performance, largely due to subtleties in coordinating access to shared data. To ensure correctness programmers often rely on conservative locking at the expense of performance. The resulting serialization of threads is a performance bottleneck. Locks also interact poorly with thread scheduling and faults, r ...

5 Fortran 8X draft

Loren P. Meissner

December 1989 **ACM SIGPLAN Fortran Forum**, Volume 8 Issue 4

Publisher: ACM Press

Full text available: pdf(21.36 MB) Additional Information: full citation, abstract, index terms

Standard Programming Language Fortran. This standard specifies the form and establishes the interpretation of programs expressed in the Fortran language. It consists of the specification of the language Fortran. No subsets are specified in this standard. The previous standard, commonly known as "FORTRAN 77", is entirely contained within this standard, known as "Fortran 8x". Therefore, any standard-conforming FORTRAN 77 program is standard conforming under this standard. New features can b ...

6 Computer Software and Copyright

Calvin N. Mooers

March 1975 **ACM Computing Surveys (CSUR)**, Volume 7 Issue 1

Publisher: ACM Press

Full text available: pdf(2.63 MB) Additional Information: full citation, references, citings, index terms

7 A first-come-first-served mutual-exclusion algorithm with small communication variables

Edward A. Lycklama, Vassos Hadzilacos

October 1991 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 13 Issue 4

Publisher: ACM Press

Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: critical selection, distributed systems, nonatomic operations

8 The family of concurrent logic programming languages

Ehud Shapiro

September 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 3

Publisher: ACM Press

Full text available: pdf(9.62 MB) Additional Information: full citation, abstract, references, citings, index terms

Concurrent logic languages are high-level programming languages for parallel and distributed systems that offer a wide range of both known and novel concurrent programming techniques. Being logic programming languages, they preserve many advantages of the abstract logic programming model, including the logical reading of programs and computations, the convenience of representing data structures with logical terms and manipulating them using unification, and the amenability to metaprogrammin ...

9 Status report of the graphic standards planning committee

Computer Graphics staff
August 1979 **ACM SIGGRAPH Computer Graphics**, Volume 13 Issue 3
Publisher: ACM Press
Full text available: pdf(15.01 MB) Additional Information: full citation, references, citings

10 Increasing availability under mutual exclusion constraints with dynamic vote reassignment

Daniel Barbara, Hector Garcia-Molina, Annemarie Spauster
November 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 4
Publisher: ACM Press
Full text available: pdf(2.53 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Voting is used commonly to enforce mutual exclusion in distributed systems. Each node is assigned a number of votes, and only the group with a majority of votes is allowed to perform a restricted operation. This paper describes techniques for dynamically reassigning votes upon node or link failure, in an attempt to make the system more resilient to future failures. We focus on autonomous methods for achieving this, that is, methods that allow the nodes to make independent choices about chan ...

11 Traits: A mechanism for fine-grained reuse

Stéphane Ducasse, Oscar Nierstrasz, Nathanael Schärli, Roel Wuyts, Andrew P. Black
March 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 28 Issue 2
Publisher: ACM Press
Full text available: pdf(1.02 MB) Additional Information: full citation, abstract, references, index terms

Inheritance is well-known and accepted as a mechanism for reuse in object-oriented languages. Unfortunately, due to the coarse granularity of inheritance, it may be difficult to decompose an application into an optimal class hierarchy that maximizes software reuse. Existing schemes based on single inheritance, multiple inheritance, or mixins, all pose numerous problems for reuse. To overcome these problems we propose *traits*, pure units of reuse consisting only of methods. We develop a for ...

Keywords: Languages, Smalltalk, inheritance, mixins, multiple inheritance, reuse, traits

12 Copyright and digital libraries

Pamela Samuelson
April 1995 **Communications of the ACM**, Volume 38 Issue 4
Publisher: ACM Press
Full text available: pdf(234.12 KB) Additional Information: full citation, abstract, references, citings, index terms

This issue of Communications highlights some of the many projects underway for the creation or enhancement of digital libraries. At the moment, no one seems to think there will be only one gargantuan digital library to sate the public's appetite for information. Rather, the expectation is that there will be many digital libraries, most of which will have specialized collections and will be internetworked together in a way loosely resembling today's Internet. Most digital li ...

13 An Elementary Discussion of Compiler/Interpreter Writing

R. L. Glass
March 1969 **ACM Computing Surveys (CSUR)**, Volume 1 Issue 1
Publisher: ACM Press
Full text available: pdf(1.85 MB) Additional Information: full citation, references, citings, index terms

14 Implicit-signal monitors

Peter A. Buhr, Ashif S. Harji
November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6
Publisher: ACM Press
Full text available: pdf(1.77 MB) Additional Information: full citation, abstract, references, index terms

An implicit (automatic) signal monitor uses a **waituntil** *predicate* statement to construct synchronization, as opposed to an explicit-signal monitor using condition variables and **signal/wait** statements for synchronization. Of the two synchronization approaches, the implicit-signal monitor is often easier to use and prove correct, but has an inherently high execution cost. Hence, its primary use is for prototyping concurrent systems using monitors, where speed and accuracy of s ...

Keywords: Automatic signal, concurrency, explicit signal, implicit signal, monitor, parallel, simulation

15 Special issue on knowledge representation

Ronald J. Brachman, Brian C. Smith
February 1980 **ACM SIGART Bulletin**, Issue 70
Publisher: ACM Press
Full text available: pdf(13.13 MB) Additional Information: full citation, abstract

In the fall of 1978 we decided to produce a special issue of the SIGART Newsletter devoted to a survey of current knowledge representation research. We felt that there were twe useful functions such an issue could serve. First, we hoped to elicit a clear picture of how people working in this subdiscipline understand knowledge representation research, to illuminate the issues on which current research is focused, and to catalogue what approaches and techniques are currently being developed. Secon ...

16 Describing and analyzing distributed software system designs

George S. Avrunin, Jack C. Wileden
July 1985 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 7 Issue 3
Publisher: ACM Press
Full text available: pdf(2.05 MB) Additional Information: full citation, abstract, references, citings, index terms, review

In this paper we outline an approach to describing and analyzing designs for distributed software systems. A descriptive notation is introduced, and analysis techniques applicable to designs expressed in that notation are presented. The usefulness of the approach is illustrated by applying it to a realistic distributed software-system design problem involving mutual exclusion in a computer network.

17 On isolation, concurrency, and the Venus rule language

Stephen Correl, Daniel P. Miranker
December 1995 **Proceedings of the fourth international conference on Information and knowledge management**
Publisher: ACM Press
Full text available: pdf(950.06 KB) Additional Information: full citation, references, citings, index terms

18 Defensive COBOL strategies

James Landon Linderman

February 1982 **ACM SIGCSE Bulletin , Proceedings of the thirteenth SIGCSE technical symposium on Computer science education SIGCSE '82**, Volume 14 Issue 1

Publisher: ACM Press

Full text available: pdf(438.44 KB) Additional Information: full citation, abstract, index terms

This paper will focus on an approach to teaching a stylistic subset of COBOL in such a manner that students are equipped with the "safe" way to program from the outset, even if this means "hiding" some of the more "dangerous" features of the language from them until they can better cope with them. This approach places the highest priority on teaching techniques which go beyond effectiveness to the broader objectives of adaptability and continued effective ...

19 An object-based programming model for shared data

Gail E. Kaiser, Brent Hailpern

April 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 14 Issue 2

Publisher: ACM Press

Full text available: pdf(3.28 MB) Additional Information: full citation, abstract, references, index terms, review

The classical object model supports private data within objects and clean interfaces between objects, and by definition does not permit sharing of data among arbitrary objects. This is a problem for real-world applications, such as advanced financial services and integrated network management, where the same data logically belong to multiple objects and may be distributed over multiple nodes on the network. Rather than give up the advantages of encapsulated objects in modeling real-world en ...

Keywords: coordination language, daemons, financial applications, object-based, real-time, sharing

20 Constraints: On context in authorization policy

Patrick McDaniel

June 2003 **Proceedings of the eighth ACM symposium on Access control models and technologies**

Publisher: ACM Press

Full text available: pdf(316.39 KB) Additional Information: full citation, abstract, references, citings, index terms

Authorization policy infrastructures are evolving with the complex environments that they support. However, the requirements and technologies supporting context are not yet well understood. Often implemented as condition functions or predefined attributes, context is used to more precisely control when and how policy is enforced. This paper considers context requirements and services in authorization policy. The properties and security requirements of context evaluation are classified. A key obs ...

Keywords: authorization, context, distributed systems, policy, policy-oriented programming, security requirements